Accepted for publication in Knowledge and Information
Systems

Collective Mining of Bayesian Networks from Distributed Heterogeneous Data

R. Chen[1], K. Sivakumar[1], and H. Kargupta[2]

[1] School of Electrical Engineering and Computer Science,
Washington State University, Pullman, WA 99164-2752, USA;
[2] Department of Computer Science and Electrical Engineering,
University of Maryland Baltimore County, Baltimore, MD 21250, USA

Abstract. We present a collective approach to learning a Bayesian network from distributed heterogenous data. In this approach, we first learn a local Bayesian network at each site using the local data. Then each site identifies the observations that are most likely to be evidence of coupling between local and non-local variables and transmits a subset of these observations to a central site. Another Bayesian network is learnt at the central site using the data transmitted from the local site. The local and central Bayesian networks are combined to obtain a collective Bayesian network, that models the entire data. Experimental results and theoretical justification that demonstrate the feasibility of our approach are presented.

Keywords: Bayesian Network, Web Log Mining, Collective Data Mining, Distributed Data Mining, Heterogenous Data

1. Introduction

Raw data is useful only when it is transformed into knowledge or useful information. This involves data analysis and transformation to extract interesting patterns and correlations among the problem variables. In practical applications, such transformations require efficient data access, analysis, and presentation of the outcome in a timely manner. For example, web server log contains records of user interactions when request for the resources in the servers is received. This contains a wealth of data for the analysis of web usage and identifying different patterns. The advent of large distributed environments in both scientific and

Received xxx
Revised xxx
Accepted xxx

Table 1. Homogeneous case: Site A with a table for credit card transaction records.

Account Number	Amount	Location	Previous record	Unusual transaction
11992346	-42.84	Seattle	Poor	Yes
12993339	2613.33	Seattle	Good	No
45633341	432.42	Portland	Okay	No
55564999	128.32	Spokane	Okay	Yes

Table 2. Homogeneous case: Site B with a table for credit card transaction records.

Account Number	Amount	Location	Previous record	Unusual transaction
87992364	446.32	Berkeley	Good	No
67845921	978.24	Orinda	Good	Yes
85621341	719.42	Walnut	Okay	No
95345998	-256.40	Francisco	Bad	Yes

commercial domains (e.g. the Internet and corporate intranets) introduces a new dimension to this process — a large number of distributed sources of data that can be used for discovering knowledge. Cost of data communication between the distributed databases is a significant factor in an increasingly mobile and connected world with a large number of distributed data sources. This cost consists of several components like (a) limited network bandwidth, (b) data security, and (c) existing organizational structure of the applications environment. The field of Distributed Knowledge Discovery and Data Mining (DDM) studies algorithms, systems, and human-computer interaction issues for knowledge discovery applications in distributed environments for minimizing this cost.

In this paper, we consider a Bayesian network (BN) model to represent uncertain knowledge. Specifically, we address the problem of learning a BN from heterogenous distributed data. It uses a collective data mining (CDM) approach introduced earlier by Kargupta et. al. (Kargupta, Huang, Sivakumar and Johnson, 2001; Kargupta, Park, Hershberger and Johnson, 2000; Hershberger and Kargupta, 2001; Park and Kargupta, 2002). Section 2 provides some background and reviews existing literature in this area. Section 3 presents the collective Bayesian learning technique. Experimental results for three datasets — one simulated and two real world — are presented in Section 4. A preliminary version of these results have been presented in (Chen, Sivakumar and Kargupta, 2001*a*; Chen, Sivakumar and Kargupta, 2001*b*). Finally, Section 5 provides some concluding remarks and directions for future work.

2. Background, Motivation, and Related Work

In this section, we provide and background and motivation to the problem by means of an example. We then review the existing literature in this area.

Distributed data mining (DDM) must deal with different possibilities of data distribution. Different sites may contain data for a common set of features of the problem domain. In case of relational data this would mean a consistent database schema across all the sites. This is the homogeneous case. Tables 1 and 2 illustrate this case using an example from a hypothetical credit card transaction domain.[1] There are two data sites A and B, connected by a network. The DDM-objective in such a domain may be to find patterns of fraudulent transactions. Note that both the tables have the same schema. The underlying distribution of the data may or may not be identical across different data sites.

In the general case the data sites may be *heterogeneous*. In other words, sites may contain tables with different schemata. Different features are observed at different sites. Let us illustrate this case with relational data. Table 3 shows two data-tables at site X. The upper table contains weather-related data and the lower one contains demographic data. Table 4 shows the content of site Y, which contains holiday toy sales data. The objective of the DDM process may be detecting relations between the toy sales, the demographic and weather related features. In the general heterogeneous case the tables may be related through different sets of key indices. For example, Tables 3 (upper) and (lower) are related through the key feature *City*; on the other hand Table 3 (lower) and Table 4 are related through key feature *State*. We consider the heterogenous data scenario in this paper.

We would like to mention that heterogenous databases, in general, could be more complicated than the above scenario. For example, there maybe a set of overlapping features that are observed at more than one site. Moreover, the existence of a key that can be used to link together observations across sites is crucial to our approach. For example, in a web log mining application, the key that can be used to link together observations across sites could be produced using either a “cookie” or the user IP address (in combination with other log data like time of access). However, these assumptions are not overly restrictive, and are required for a reasonable solution to the distributed Bayesian learning problem.

2.1. Motivation

Bayesian networks offer very useful information about the mutual dependencies among the features in the application domain. Such information can be used for gaining better understanding about the dynamics of the process under observation. Financial data analysis, manufacturing process monitoring, sensor data analysis, web mining are a few examples where mining Bayesian networks has been quite useful. Bayesian techniques will also be useful for mining distributed data. In this section we discuss examples of such scenario and explain how the proposed collective Bayesian learning algorithm can be useful in practice.

Advances in computing and communication over wired and wireless networks have resulted in many pervasive distributed computing environments. The Internet, intranets, local area networks, and ad hoc wireless networks are some examples. Many of these environments have different distributed sources of voluminous data and multiple compute nodes. Since data mining offers the capability of sifting through data in search of useful information, it finds many applications in distributed systems just like their centralized monolithic counterparts.

[1] Please note that the credit card domain may not always have consistent schema. The domain is used just for illustration.

Table 3. Heterogeneous case: Site X with two tables, one for weather and the other for demography.

City	Temp.	Humidity	Wind Chill
Boise	20	24%	10
Spokane	32	48%	12
Seattle	63	88%	4
Portland	51	86%	4
Vancouver	47	52%	6

City	State	Size	Average earning	Proportion of small businesses
Boise	ID	Small	Low	0.041
Spokane	WA	Medium	Medium	0.022
Seattle	WA	Large	High	0.014
Portland	OR	Large	High	0.017
Vancouver	BC	Medium	Medium	0.031

Table 4. Heterogeneous case: Site Y with one table holiday toy sales.

State	Best Selling Item	Price ($)	Number Items Sold (In thousands)
WA	Snarc Action Figure	47.99	23
ID	Power Toads	23.50	2
BC	Light Saber	19.99	5
OR	Super Squirter	24.99	142
CA	Super Fun Ball	9.99	24

There are many domains where distributed processing of data is a more natural and scalable solution. For example, consider an ad hoc wireless sensor network where the different sensor nodes are monitoring some time-critical events. Central collection of data from every sensor node may create heavy traffic over the limited bandwidth wireless channels and this may also drain a lot of power from the devices. A distributed architecture for data mining is likely to reduce the communication load and also reduce the battery power more evenly across the different nodes in the sensor network. One can easily imagine similar needs for distributed computation of data mining primitives in ad hoc wireless networks of mobile devices like PDAs, cellphones, and wearable computers. Potential applications include personalization, collaborative process monitoring, intrusion detection over ad hoc wireless networks. We need data mining architectures that pay careful attention to the distributed resources of data, computing, and communication in order to consume them in a near optimal fashion. Distributed data mining (DDM) considers data mining in this broader context. The objective of DDM is to perform the data mining operations based on the type and availability of the distributed resources. It may choose to download the data sets to a single site and perform the data mining operations at a central location. However, that decision in DDM should be based on the properties of the computing, storage, and communication capabilities. This is in contrast with the traditional centralized data mining methodology where collection of data at a single location prior to analysis is an invariant characteristic.

The wireless domain is not the only example. In fact, most of the applications that deal with time-critical distributed data are likely to benefit by paying careful attention to the distributed resources for computation, storage, and the cost of communication. The world wide web is a very good example. It contains distributed data and computing resources. An increasing number of databases (e.g. weather databases, oceanographic data at www.noaa.gov), and data streams (e.g. financial data at www.nasdaq.com, emerging disease information at www.cdc.gov) are coming online; many of them change frequently. It is easy to think of many applications that require regular monitoring of these diverse and distributed sources of data. A distributed approach to analyze this data is likely to be more scalable and practical particularly when the application involves a large number of data sites. The distributed approach may also find applications in mining remote sensing and astronomy data. For example, the NASA Earth Observing System (EOS), a data collector for a number of satellites, holds 1450 data sets that are stored, managed, and distributed by the different EOS Data and Information System (EOSDIS) sites that are geographically located all over the USA. A pair of Terra spacecraft and Landsat 7 alone produces about 350 GB of EOSDIS data per day. An online mining system for EOS data streams may not scale if we use a centralized data mining architecture. Mining the distributed EOS repositories and associating the information with other existing environmental databases may benefit from DDM. In astronomy, the size of telescope image archives have already reached the terabyte range and they continue to increase very fast as information is collected for new all-sky surveyors such as the GSC-II (McLean, Hawkins, Spagna, Lattanzi, Lasker, Jenkner and White, 1998) and the Sloan Digital Survey (Szalay, 1998). DDM may offer a practical scalable solution for mining these large distributed astronomy data repositories.

DDM may also be useful in environments with multiple compute nodes connected over high speed networks. Even if the data can be quickly centralized using the relatively fast network, proper balancing of computational load among a cluster of nodes may require a distributed approach.

2.2. Related Work

We now review important literature on BN learning. A BN is a probabilistic graphical model that represents uncertain knowledge (Jensen, 1996). Spiegelhalter and Lauritzen (1990) and Buntine (1991) discuss parameter learning of a BN from complete data, whereas Binder, Koller, Russel and Kanazawa (1997) and Thiesson (1995) discuss parameter learning from incomplete data using gradient method. Lauritzen (1995) has proposed an EM algorithm to learn Bayesian network parameters, whereas Bauer, Koller and Singer (1997) describe methods for accelerating convergence of the EM algorithm. Learning using Gibbs sampling has been proposed by Thomas, Spiegelhalter and Gilks (1992) and Gilks, Richardson and Spiegelhalter (1996). The Bayesian score to learn the structure of a BN is discussed by Cooper and Herskovits (1992), Buntine (1991), and Heckerman, Geiger and Chickering (1995). Learning the structure of a BN based on the Minimal Description Length (MDL) principle has been presented by Bouckaert (1994), Lam and Bacchus (1994), and Suzuki (1993). Learning BN structure using greedy hill-climbing and other variants was introduced by Heckerman and Gieger (1995), whereas Chickering (1996) introduced a method based on search over equivalence network classes. Methods for approximating full Bayesian model averaging were presented by Buntine (1991), Heckerman and Gieger (1995), and Madigan and Raftery (1994).

Learning the structure of BN from incomplete data was considered by Chickering and Heckerman (1997), Cheeseman and Stutz (1996), Friedman (1998), Meila and Jordan (1998), and Singh (1997). The relationship between causality and Bayesian networks has been discussed by Heckerman and Gieger (1995), Pearl (1993), and Spirtes, Glymour and Scheines (1993). Buntine (1991), Friedman and Goldszmidt (1997), and Lam and Bacchus (1994) discuss how to sequentially update the structure of a BN based on additional data. Applications of Bayesian network to clustering (AutoClass) and classification has been presented in (Cheeseman and Stutz, 1996; Ezawa and T, 1995; Friedman, Geiger and Goldszmidt, 1997; Singh and Provan, 1995). Zweig and Russel (1998) have used BNs for speech recognition, whereas Breese, Heckerman and Kadie (1998) have discussed collaborative filtering methods that use BN learning algorithms. Applications to causal learning in social sciences has been presented by Spirtes et al. (1993)

An important problem is how to learn the Bayesian network from data in distributed sites. The centralized solution to this problem is to download all datasets from distributed sites. Kenji (1997) has worked on the homogeneous distributed learning scenario. In this case, every distributed site has the same feature but different observations. In this paper, we address the heterogenous case, where each site has data about only a subset of the features. To our knowledge, there is no significant work that addresses the heterogenous case.

3. Collective Bayesian Learning

In the following, we briefly review Bayesian networks and then discuss our collective approach to learning a Bayesian network that is specifically designed for a distributed data scenario.

3.1. Bayesian Networks: A review

A Bayesian network (BN) is a probabilistic graph model. It can be defined as a pair ($\mathcal{G}$, p), where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed acyclic graph (DAG) (Jensen, 1996; Heckerman, 1998). Here, $\mathcal{V}$ is the vertex set which represents variables in the problem and $\mathcal{E}$ is the edge set which denotes probabilistic relationships among the variables. For a variable $X \in \mathcal{V}$, a parent of X is a node from which there is a directed link to X. Let $pa(X)$ denote the set of parents of X, then the conditional independence property can be represented as follows:

$$P(X \mid \mathcal{V} \setminus X) = P(X \mid pa(X)). \tag{1}$$

This property can simplify the computations in a Bayesian network model. For example, the joint distribution of the set of all variables in $\mathcal{V}$ can be written as a product of conditional probabilities as follows:

$$P(\mathcal{V}) = \prod_{X \in \mathcal{V}} P(X \mid pa(X)). \tag{2}$$

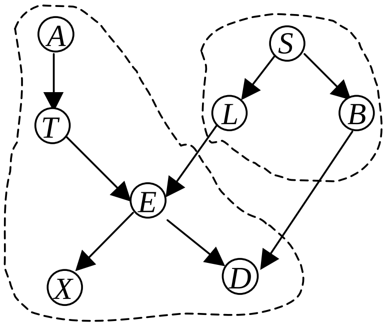

Fig. 1. ASIA Model

The conditional independence between variables is either obtained from a priori expert knowledge or discerned from data, or a combination of both (Jensen, 1996). The set of conditional distributions $\{P(X \mid pa(X)), X \in \mathcal{V}\}$ are called the parameters of a Bayesian network. Note that if variable X has no parents, then $P(X \mid pa(X)) = P(X)$ is the marginal distribution of X.

Figure 1 is a Bayesian network called the ASIA model (adapted from Lauritzen and Spiegelhalter (1988)). The variables are Dyspnoea, Tuberculosis, Lung cancer, Bronchitis, Asia, X-ray, Either, and Smoking. They are all binary variables. The joint distribution of all variables is

$$\begin{aligned} P(A,S,T,L,B,E,X,D) = {} & P(A)P(S)P(T \mid A)P(L \mid S)P(B \mid S) \\ & P(E \mid T,L)P(X \mid E)P(D \mid B,E). \end{aligned} \tag{3}$$

The ordering of variables constitutes a constraint on the structure of a Bayesian network. If variable X appears before variable Y, then Y can not be a parent of X. We use the ordering (A,S,T,L,B,E,X,D) as prior knowledge in our example.

Two important issues in using a Bayesian network are : (a) learning a Bayesian network and (b) probabilistic inferencing. Learning a BN involves learning the structure of the network (the directed graph), and obtaining the conditional probabilities (parameters) associated with the network. Once a Bayesian network is constructed, we usually need to determine various probabilities of interest from the model. This process is referred to as probabilistic inference. For example, in the ASIA model, a diagnosis application would require finding the probability $P(B \mid D)$ of Bronchitis, given the (observed) symptom Dyspnoea. This probability (usually called posterior probability) can be computed using the Bayes rule.

3.2. Collective Bayesian Network Learning Strategy

We now present a collective strategy to learn a Bayesian network (both structure and parameters) when data is distributed among different sites. The centralized solution to this problem is to download all datasets from distributed sites to a central site. In many applications, this would not be feasible because of the size of the data, available communication bandwidth, or due to security considerations. Learning a BN for the homogeneous case was studied by Kenji (1997). In this case, every distributed site has the same set of features but has different set of observations. We address here the heterogenous case, where each distributed site has all the observations for only a subset of the features.

The primary steps in our approach are:

- Compute local BNs (local model) involving the variables observed at each site (local variables) based on local data.
- At each site, based on the local BN, identify the observations that are most likely to be evidence of coupling between local and non-local variables. Transmit a subset of these observations to a central site.
- At the central site, a limited number of observations of all the variables are now available. Using this, compute a non-local BN consisting of links between variables across two or more sites.
- Combine the local models with the links discovered at the central site to obtain a collective BN.

The non-local BN thus constructed would be effective in identifying associations between variables across sites, whereas the local BNs would detect associations among local variables at each site. The conditional probabilities can also be estimated in a similar manner. Those probabilities that involve only variables from a single site can be estimated locally, whereas the ones that involve variables from different sites can be estimated at the central site. The same methodology could be used to update the network based on new data. First, the new data is tested for how well it fits with the local model. If there is an acceptable statistical fit, the observation is used to update the local conditional probability estimates. Otherwise, it is also transmitted to the central site to update the appropriate conditional probabilities (of cross terms). Finally, a collective BN can be obtained by taking the union of nodes and edges of the local BNs and the nonlocal BN, along with the conditional probabilities from the appropriate BNs. Probabilistic inference can now be performed based on this collective BN. Note that transmitting the local BNs to the central site would involve a significantly lower communication as compared to transmitting the local data.

It is quite evident that learning probabilistic relationships between variables that belong to a single local site is straightforward and does not pose any additional difficulty as compared to a centralized approach.[2] The important objective is to correctly identify the coupling between variables that belong to two (or more) sites. These correspond to the edges in the graph that connect variables between two sites and the conditional probability(ies) at the associated node(s). In the following, we describe our approach to selecting observations at the local sites that are most likely to be evidence of strong coupling between variables at two different sites.

3.3. Selection of samples for transmission to global site

For simplicity, we will assume that the data is distributed between two sites and will illustrate the approach using the BN in Figure 1. The extension of this approach to more than two sites is straightforward. Let us denote by $\mathcal{A}$ and $\mathcal{B}$ the variables in the left and right groups, respectively, in Figure 1. We assume that

[2] This may not be true for arbitrary Bayesian network structure. We will discuss this issue further in the last section.

the observations for $\mathcal{A}$ are available at site A, whereas the observations for $\mathcal{B}$ are available at a different site B. Furthermore, we assume that there is a common feature ("key" or index) that can be used to associate a given observation in site A to a corresponding observation in site B. Naturally, $\mathcal{V} = \mathcal{A} \cup \mathcal{B}$.

At each local site, a local Bayesian network can be learned using only samples in this site. This would give a BN structure involving only the local variables at each site and the associated conditional probabilities. Let $p_A(.)$ and $p_B(.)$ denote the estimated probability function involving the local variables. This is the product of the conditional probabilities as indicated by (2). Since $p_A(x)$, $p_B(x)$ denote the probability or likelihood of obtaining observation x at sites A and B, we would call these probability functions the likelihood functions $l_A(.)$ and $l_B(.)$, for the local model obtained at sites A and B, respectively. The observations at each site are ranked based on how well it fits the local model, using the local likelihood functions. The observations at site A with large likelihood under $l_A(.)$ are evidence of "local relationships" between site A variables, whereas those with low likelihoods under $l_A(.)$ are possible evidence of "cross relationships" between variables across sites. Let $S(A)$ denote the set of keys associated with the latter observations (those with low likelihood under $l_A(.)$). In practice, this step can be implemented in different ways. For example, we can set a threshold ρ_A and if $l_A(x) \leq \rho_A$, then $x \in S_A$. The sites A and B transmit the set of keys S_A, S_B, respectively, to a central site, where the intersection $S = S_A \cap S_B$ is computed. The observations corresponding to the set of keys in S are then obtained from each of the local sites by the central site.

The following argument justifies our selection strategy. Using the rules of probability, and the assumed conditional independence in the BN of Figure 1, it is easy to show that:

$$P(\mathcal{V}) = P(\mathcal{A}, \mathcal{B}) = P(\mathcal{A} \mid \mathcal{B})P(\mathcal{B}) = P(\mathcal{A} \mid nb(\mathcal{A}))P(\mathcal{B}), \tag{4}$$

where $nb(\mathcal{A}) = \{B, L\}$ is the set of variables in $\mathcal{B}$, which have a link connecting it to a variable in $\mathcal{A}$. In particular,

$$P(\mathcal{A} \mid nb(\mathcal{A})) = P(A)P(T \mid A)P(X \mid E)P(E \mid T, L)P(D \mid E, B). \tag{5}$$

Note that, the first three terms in the right-hand side of (5) involve variables local to site A, whereas the last two terms are the so-called *cross terms*, involving variables from sites A and B. Similarly, it can be shown that

$$P(\mathcal{V}) = P(\mathcal{A}, \mathcal{B}) = P(\mathcal{B} \mid \mathcal{A})P(\mathcal{A}) = P(\mathcal{B} \mid nb(\mathcal{B}))P(\mathcal{A}), \tag{6}$$

where $nb(\mathcal{B}) = \{E, D\}$ and

$$P(\mathcal{B} \mid nb(\mathcal{B})) = P(S)P(B \mid S)P(L \mid S)P(E \mid T, L)P(D \mid E, B). \tag{7}$$

Therefore, an observation $\{A = a, T = t, E = e, X = x, D = d, S = s, L = l, B = b\}$ with low likelihood at both sites A and B; i.e. for which both $P(\mathcal{A})$ and $P(\mathcal{B})$ are small, is an indication that both $P(\mathcal{A} \mid nb(\mathcal{A}))$ and $P(\mathcal{B} \mid nb(\mathcal{B}))$ are large for that observation (since observations with small $P(\mathcal{V})$ are less likely to occur). Notice from (5) and (7) that the terms common to both $P(\mathcal{A} \mid nb(\mathcal{A}))$ and $P(\mathcal{B} \mid nb(\mathcal{B}))$ are precisely the conditional probabilities that involve variables from both sites A and B. In other words, this is an observation that indicates a coupling of variables between sites A and B and should hence be transmitted to a central site to identify the specific coupling links and the associated conditional probabilities.

In a sense, our approach to learning the cross terms in the BN involves a selective sampling of the given dataset that is most relevant to the identification of coupling between the sites. This is a type of *importance sampling*, where we select the observations that have high conditional probabilities corresponding to the terms involving variables from both sites. Naturally, when the values of the different variables (features) from the different sites, corresponding to these selected observations are pooled together at the central site, we can learn the coupling links as well as estimate the associated conditional distributions. These selected observations will, by design, not be useful to identify the links in the BN that are local to the individual sites. This has been verified in our experiments (see Section 4).

3.4. Performance Analysis

In the following, we present a brief theoretical analysis of the performance of the proposed collective learning method. We compare the performance of our collective BN with that of a Bayesian network learned using a centralized approach (referred to as centralized BN in the sequel).

There are two types of errors involved in learning a BN: (a) Error in BN structure and (b) Error in parameters (probabilities) of the BN. The structure error is defined as the sum of the number of correct edges missed and the number of incorrect edges detected. For parameter error, we need to quantify the "distance" between two probability distributions. We only consider learning error in the parameters, assuming that the structure of the BN has been correctly determined (or is given). A widely used metric is the Kullback-Leibler (KL) distance (cross-entropy measure) $d_{KL}(p,q)$ between two discrete probabilities, $\{p_i\}$, $\{q_i\}$, $i = 1, 2, \ldots, N$

$$d_{KL}(p,q) = \sum_{i=1}^{N} p_i \ln(\frac{p_i}{q_i}) \tag{8}$$

where N is the number of possible outcomes.

Indeed, if p^* is the empirically observed distribution for data samples $\{s_i, 1 \leq i \leq M\}$ and h is a hypothesis (candidate probability distribution for the underlying true distribution), then (Abe, Takeuchi and Warmuth, 1991)

$$\begin{aligned} d_{KL}(p^*,h) &= \sum_{i=1}^{M} p^*(s_i) \ln(\frac{p^*(s_i)}{h(s_i)}) = \sum_{i=1}^{M} \frac{1}{M} \ln \frac{1}{M} - \sum_{i=1}^{M} \frac{1}{M} \ln(h(s_i)) \\ &= \ln \frac{1}{M} - \frac{1}{M} \sum_{i=1}^{M} \ln(h(s_i)). \end{aligned} \tag{9}$$

Therefore, minimizing the KL distance with respect to the empirically observed distribution is equivalent to finding the maximum likelihood solution h^* of $\sum_{i=1}^{M} \ln(h(s_i))$.

Since the BN provides a natural factorization of the joint probability in terms of the conditional probabilities at each node (see (2)), it is convenient to express the KL distance between two joint distributions in terms of the corresponding conditional distributions. Let h and c be two possible (joint) distributions of the variables in a BN. For $i = 1, 2, \ldots, n$, let $h_i(x_i \mid \pi_i)$, $c_i(x_i \mid \pi_i)$ be the corresponding conditional distribution at node i, where x_i is the variable at node i and π_i is the set of parents of node i. Following Dasgupta (1997), define a distance $d_{CP}(P, c_i, h_i)$ between h_i and c_i with respect to the true distribution P:

$$d_{CP}(P, c_i, h_i) = \sum_{\pi_i} P(\pi_i) \sum_{x_i} P(x_i \mid \pi_i) \ln(\frac{c_i(x_i \mid \pi_i)}{h_i(x_i \mid \pi_i)}). \tag{10}$$

It is then easy to show that

$$d_{KL}(P, h) - d_{KL}(P, c) = \sum_{i=1}^{n} d_{CP}(P, c_i, h_i). \tag{11}$$

Equations (10) and (11) provide a useful decomposition of the KL distance between the true distribution P and two different hypotheses c, h. This will be useful in our analysis of sample complexity in the following sub-section.

3.5. Sample Complexity

We now derive a relationship between the accuracy of collective BN and the number of samples transmitted to the central site. We consider the unrestricted multinomial class BN, where all the node variables are Boolean. The hypothesis class H is determined by the set of possible conditional distributions for the different nodes. Given a BN of n variables and a hypothesis class H, we need to choose a hypothesis $h \in H$ which is close to a unknown distribution P. Given an error threshold ϵ and a confidence threshold δ, we are interested in constructing a function $N(\epsilon, \delta)$, such that if the number of samples M is larger than $N(\epsilon, \delta)$

$$Prob(d_{KL}(P, h) < d_{KL}(P, h_{opt}) + \epsilon) > 1 - \delta, \tag{12}$$

where $h_{opt} \in H$ is the hypothesis that minimizes $d_{KL}(P, h)$. If smallest value of $N(\epsilon, \delta)$ that satisfies this requirement is called the sample complexity. This is usually referred to as the probably approximately correct (PAC) framework. Friedman and Yakhini (1996) have examined the sample complexity of the maximum description length principle (MDL) based learning procedure for BNs.

Dasgupta (1997) gave a thorough analysis for the multinomial model with Boolean variables. Suppose the BN has n nodes and each node has at most k parents. Given ϵ and δ, an upper bound of sample complexity is

$$N(\epsilon, \delta) = \frac{288n^2 2^k}{\epsilon^2} \ln^2(1 + \frac{3n}{\epsilon} \ln \frac{18n^2 2^k \ln(1 + 3n/\epsilon)}{\epsilon\delta}). \tag{13}$$

Equation (13) gives a relation between the sample size and the (ϵ, δ) bound. For the conditional probability $h_i(x_i \mid \pi_i) = P(X_i = x_i \mid \Pi_i = \pi_i)$, we have (see (10))

$$d_{CP}(P, h_{opt}, h) \leq \frac{\epsilon}{n} \tag{14}$$

We now use the above ideas to compare the performance of the collective learning method with the centralized method. We fix the confidence δ and suppose that an ϵ^{cen} can be found for the centralized method, for a given sample size M using (13). Then, following the analysis by Dasgupta (1997),

$$d_{CP}(P, h_{opt}^{cen}, h^{cen}) \leq \frac{\epsilon^{cen}}{n}, \tag{15}$$

where h_{opt}^{cen} is the optimal hypothesis and h^{cen} is the hypothesis obtained based on a centralized approach. Then from (11)

$$d_{KL}(P, h^{cen}) - d_{KL}(P, h_{opt}^{cen}) = \sum_{i=1}^{n} d_{CP}(P, h_{i,opt}^{cen}, h_i^{cen}) \leq \sum_{i=1}^{n} \frac{\epsilon^{cen}}{n} = \epsilon^{cen}. \quad (16)$$

For the collective BN learning method, the set of nodes can be split into two parts. Let V_l be the set of nodes, which have all their parent nodes at the same local site, and V_c be the set of nodes, which have at least one parent node belonging to a site different than the node itself. For ASIA model, $V_l = \{A, S, T, L, B, X\}$ and $V_c = \{E, D\}$. We use n_l and n_c to denote the cardinality of the sets V_l and V_c. If a node $x \in V_l$, the collective method can learn the conditional probability $P(x \mid pa(x))$ using all data because this depends only on the local variables. Therefore, for $x \in V_l$,

$$d_{CP}(P, h_{opt}^{col}, h^{col}) \leq \frac{\epsilon_1^{col}}{n} = \frac{\epsilon^{cen}}{n}, \quad (17)$$

where, for the local terms, $\epsilon_1^{col} = \epsilon^{cen}$. For the nodes in V_c, only the data transmitted to the central site can be used to learn its conditional probability. Suppose M_c data samples are transmitted to the central site, and the error threshold ϵ_2^{col} satisfies (13), for the same fixed confidence $1 - \delta$. Therefore, for $x \in V_c$, we have from (14) that $d_{CP}(P, h_{opt}^{col}, h^{col}) \leq \frac{\epsilon_2^{col}}{n}$, where $\epsilon_2^{col} \geq \epsilon^{cen}$, in general, since the in the collective learning method, only $M_c \leq M$ samples are available at the central site. Then from (11) and (17)

$$\begin{aligned} d_{KL}(P, h_{opt}^{col}) - d_{KL}(P, h^{col}) &= \sum_{i=1}^{n} d_{CP}(P, h_{i,opt}^{col}, h_i^{col}) \\ &= \sum_{i \in V_l} d_{CP}(P, h_{i,opt}^{col}, h_i^{col}) + \sum_{i \in V_c} d_{CP}(P, h_{i,opt}^{col}, h_i^{col}) \\ &= \frac{n_l}{n}\epsilon^{cen} + \frac{n_c}{n}\epsilon_2^{col} \end{aligned} \quad (18)$$

Comparing (16) and (18), it is easy to see that the error threshold of the collective method is $\epsilon^{col} = \frac{n_l}{n}\epsilon^{cen} + \frac{n_c}{n}\epsilon_2^{col}$. The difference of the error threshold between the collective and the centralized method is

$$\epsilon^{col} - \epsilon^{cen} = \frac{n_c}{n}(\epsilon_2^{col} - \epsilon^{cen}) \quad (19)$$

Equation (19) shows two important properties of the collective method. First, the difference in performance is independent of the variables in V_l. This means the performance of the collective method for the parameters of local variables is same as that of the centralized method. Second, the collective method is a tradeoff between accuracy and the communication overhead. The more data we communicate, more closely ϵ_2^{col} will be to ϵ^{cen}. When $M_c = M$, $\epsilon_2^{col} = \epsilon^{cen}$, and $\epsilon^{col} - \epsilon^{cen} = 0$.

Server	Client
selecting samples for transmission detecting cross-terms parameter learning for cross-terms collective BN assembling data transmission control	structure learning of local term parameter learning of local term likelihood computation

Table 5. Functionality of client/server in DistrBN

4. Experimental Results

We tested our approach on three different datasets — ASIA model, real web log data, and ALARM network. We present our results for the three cases in the following subsections. A software package called DistrBN has been developed to implement our proposed algorithm. The software is based on a BN C++ library called SMILE[3]. It has a client/server architecture, where a client program runs at each local site. The server can be at a central site or one of the local sites. The functionality of client/server is given in Table 5. When we start a distributed BN learning task using DistrBN, the clients in local sites take charge of the local BN learning and likelihood computing. Based on the indices of the low likelihood samples at each site, the server determines the indices of samples needed for detecting cross-links. The samples corresponding to these indices are the transmitted from each of the local sites to the server, along with the local BN model. The server learns the cross-terms and incorporates the local models along with the cross-terms to obtain a collective BN.

4.1. ASIA Model

This experiment illustrates the ability of the proposed collective learning approach to correctly obtain the structure of the BN (including the cross-links) as well as the parameters of the BN. Our experiments were performed on a dataset that was generated from the BN depicted in Figure 1 (ASIA Model). The conditional probability of a variable is a multidimensional array, where the dimensions are arranged in the same order as ordering of the variables, viz. $\{A, S, T, L, B, E, X, D\}$. Table 6 (top) depicts the conditional probability of node E. It is laid out such that the first dimension toggles fastest. From Table 6, we can write the conditional probability of node E as a single vector as follows: $[0.9, 0.1, 0.1, 0.01, 0.1, 0.9, 0.9, 0.99]$. The conditional probabilities (parameters) of ASIA model are given in Table 6 (bottom) following this ordering scheme. We generated $n = 6000$ observations from this model, which were split into two sites as illustrated in Figure 1 (site A with variables A, T, E, X, D and site B with variables S, L, B). The split of variables into two sites was arbitrary and done in such a fashion to yield two cross-terms. In practice, we do not have control over the distribution of variables among different sites — this is dictated by what variables are observed (and stored) at each site. Note that there are two edges ($L \rightarrow E$ and $B \rightarrow D$) that connect variables from site A to site B, the rest of the six edges being local.

[3] homepage: `http://www2.sis.pitt.edu/~genie/`

No.	T	L	E	Probability
1	F	F	F	0.9
2	T	F	F	0.1
3	F	T	F	0.1
4	T	T	F	0.01
5	F	F	T	0.1
6	T	F	T	0.9
7	F	T	T	0.9
8	T	T	T	0.99

A	0.99	0.01						
S	0.5	0.5						
T	0.1	0.9	0.9	0.1				
L	0.3	0.6	0.7	0.4				
B	0.1	0.8	0.9	0.2				
E	0.9	0.1	0.1	0.01	0.1	0.9	0.9	0.99
X	0.2	0.6	0.8	0.4				
D	0.9	0.1	0.1	0.01	0.1	0.9	0.9	0.99

Table 6. (Top) The conditional probability of node E and (Bottom) All conditional probabilities for the ASIA model

Local A				
A	0.99	0.01		
T	0.10	0.84	0.90	0.16
E	0.50	0.05	0.50	0.95
X	0.20	0.60	0.80	0.40
D	0.55	0.05	0.45	0.95
Local B				
S	0.49	0.51		
L	0.30	0.59	0.70	0.41
B	0.10	0.81	0.90	0.19

Table 7. The conditional probabilities of local site A and local site B

Local Bayesian networks were constructed using a conditional independence test based algorithm (Cheng, Bell and Liu, 1997), for learning the BN structure and a maximum likelihood based method for estimating the conditional probabilities. The local networks were exact as far as the edges involving only the local variables. We then tested the ability of the collective approach to detect the two non-local edges. The estimated parameters of these two local Bayesian network is depicted in Table 7. Clearly, the estimated probabilities at all nodes, except nodes E and D, are close to the true probabilities given in Table 6. In other words, the parameters that involve only local variables have been successfully learnt at the local sites.

A fraction of the samples, whose likelihood are smaller than a selected threshold T, were identified at each site. In our experiments, we set

$$T_i = \mu_i + \alpha\sigma_i, \quad i \in \{A, B\}, \tag{20}$$

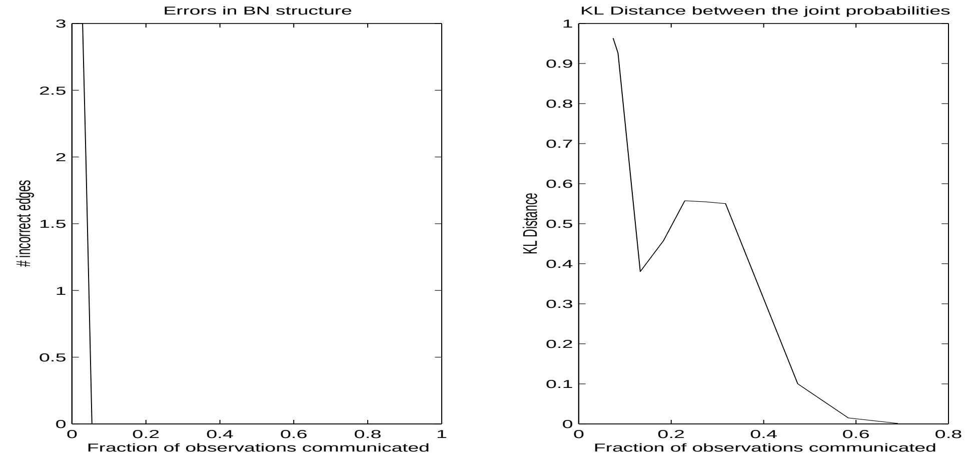

Fig. 2. Performance of collective BN: (left) structure learning error (right) parameter learning error.

for some constant α, where μ_i is the (empirical) mean of the local likelihood values and σ_i is the (empirical) standard deviation of the local likelihood values. The samples with likelihood less than the threshold (T_A at site A T_B at site B) at both sites were sent to a central site. The central site learns a global BN based on these samples. Finally, a collective BN is formed by taking the union of edges detected locally and those detected at the central site. The error in structure learning of the collective Bayesian network is defined as the sum of the number of correct edges missed and the number of incorrect edges detected. This is done for different values of α. Figure 2 (left) depicts this error as a function of the number of samples communicated (which is determined by α). It is clear that the exact structure can be obtained by transmitting about 5% of the total samples.

Next we assessed the accuracy of the estimated conditional probabilities. For the collective BN, we used the conditional probabilities from local BN for the local terms and the ones estimated at the global site for the cross terms. This was compared with the performance of a BN learnt using a centralized approach (by aggregating all data at a single site). Figure 2 (right) depicts the KL distance $D(p_{cntr}(\mathcal{V}), p_{coll}(\mathcal{V}))$ between the joint probabilities computed using our collective approach and the one computed using a centralized approach. Clearly, even with a small communication overhead, the estimated conditional probabilities based on our collective approach is quite close to that obtained from a centralized approach.

A more important test of our approach is the error in estimating the conditional probabilities of the cross terms, estimated at the global site, based on a selective subset of data. In this paper, a metric called conditional KL (CKL) distance is defined as follows:

$$D_{CKL}(i, B_{cntr}, B_{coll}) = \sum_j p_{cntr}(j) \cdot D_{KL}(p^{ij}_{cntr}, p^{ij}_{coll}). \tag{21}$$

$D_{CKL}(i, B_{cntr}, B_{coll})$ is the distance between two conditional probability tables of node x_i. Note that each row of CPT is a distribution with fixed parent configuration $pa(i) = j$. So $D_{KL}(p^{ij}_{cntr}, p^{ij}_{coll})$ is the KL distance of variable x_i with a

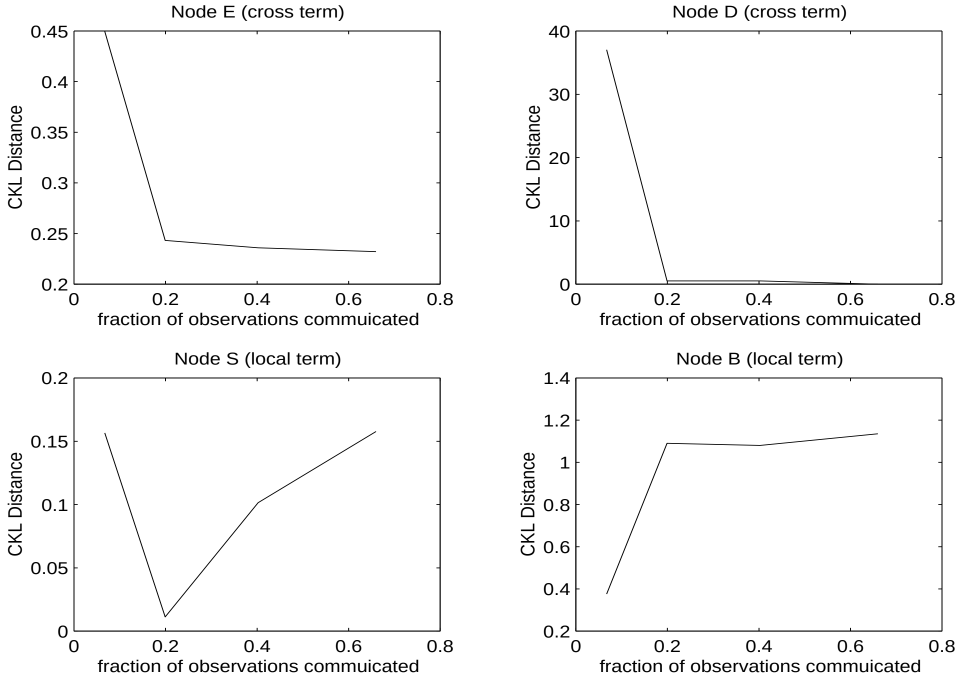

Fig. 3. KL distance between conditional probabilities

specific parent configuration j. CKL distance is derived from equation 10. True distribution P in equation 10 is replaced by p_{cntr} since we can not get P in real world application. The KL distance between the conditional probabilities is computed based on our collective BN and a BN obtained using a centralized approach (by transmitting all data to one site), for the cross terms: $p(E \mid T, L)$ and $p(D \mid E, B)$. Figure 3 (top left) depicts the CKL distance of node E and figure 3 (top right) depicts the CKL distance of node D. Clearly, even with a small data communication, the estimates of the conditional probabilities of the cross-terms, based on our collective approach, is quite close to that obtained by the centralized approach. To further verify the validity of our approach, the transmitted data at the central site was used to estimate two local terms, node S and node B. The corresponding CKL distances are depicted in the bottom row of Figure 3 (left: node S and right: node B). It is clear that the estimates of these probabilities is quite poor. This clearly demonstrates that our technique can be used to perform a biased sampling for discovering relationships between variables across sites.

4.2. Webserver Log Data

In the second set of experiments, we used data from a real world domain — a web server log data. This experiment illustrates the ability of the proposed collective learning approach to learn the parameters of a BN from real world web log data. A web server log contains records of user interactions when a request for the resources in the servers is received. Web log mining can provide useful information about different user profiles. This in turn can be used to offer personalized services as well as to better design and organize the web resources based on usage history.

In our application, the raw web log file was obtained from the web server of the School of EECS at Washington State University — `http://www.eecs.wsu.edu`. There are three steps in our processing. First we preprocess the raw web log file to transform it to a session form which is useful to our application. Before we do the transform, we should clean the web log file. There are many redundant records in the web log file. For example, a page may contain several pictures. So when someone accesses this page, the web log file will have records for accessing these pictures. But only the record for accessing that page is enough so we remove all other records. This step can reduce the web log file size dramatically. Then we transform the web log file to a session form. This involves identifying a sequence of logs as a single session, based on the IP address (or cookies if available) and time of access. Each session corresponds to the logs from a single user in a single web session. We consider each session as a data sample. Then we categorize the resource (html, video, audio etc.) requested from the server into eight categories: E-Admission, F-Course, G-EECS Home, and H-Research. E-EE Faculty, C-CS Faculty, L-Lab and facilities, T-Contact Information, A-Admission Information, U-Course Information, H-EECS Home, and R-Research. These categories are our features. In general, we would have several tens (or perhaps a couple of hundred) of categories, depending on the webserver. This categorization has to be done carefully, and would have to be automated for a large web server. Finally, Each feature value in a session is set to one or zero, depending on whether the user requested resources corresponding to that category. An 8-feature, binary dataset was thus obtained, which was used to learn a BN. Figure 4 illustrates this process schematically.

A central BN was first obtained using the whole dataset. Figure 5 depicts the structure of this centralized BN. We then split the features into two sets, corresponding to a scenario where the resources are split into two different web servers. Site A has features E, C, T, and U and site B has features L, A, H, and R. We assumed that the BN structure was known, and estimated the parameters (probability distribution) of the BN using our collective BN learning approach. Figure 6 shows the KL distance between the central BN and the collective BN as a function of the fraction of observations communicated. Clearly the parameters of collective BN is close to that of central BN even with a small fraction of data communication.

4.3. ALARM Network

This experiment illustrates the scalability of our approach with respect to number of (a) sites, (b) features, and (c) observations. We test only the scalability of BN parameter learning, assuming that the network structure is given. We used a real world BN application called ALARM network, which has 37 nodes and 46 edges. It is a successful application of BN in the medical diagnosis area. ALARM network is a widely used benchmark network to evaluate the algorithm. The ALARM network has been developed for on-line monitoring of patients in intensive care units and generously contributed to the community by Beinlich, Suermondt, Chavez and Cooper (1989). The structure of ALARM network

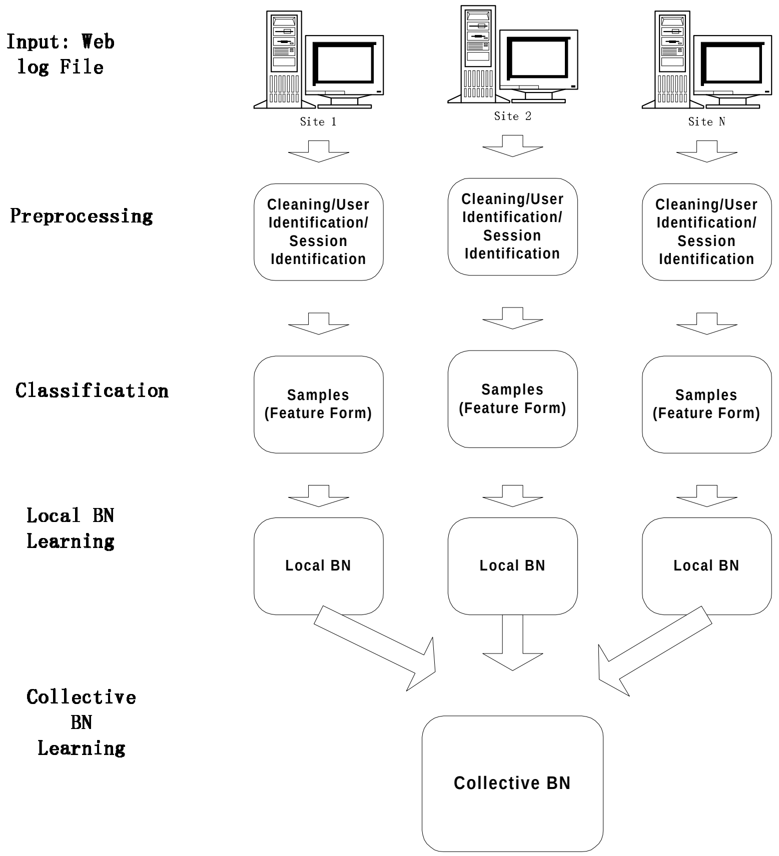

Fig. 4. Schematic illustrating preprocessing and mining of web log data

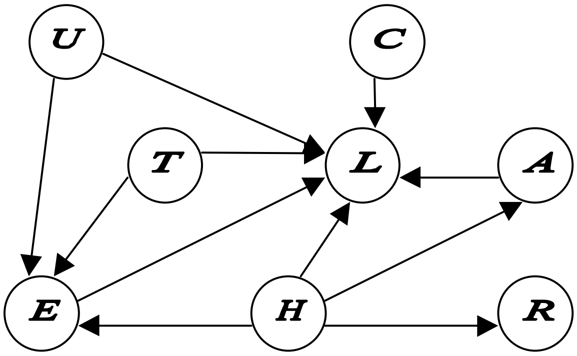

Fig. 5. Web log mining BN structure

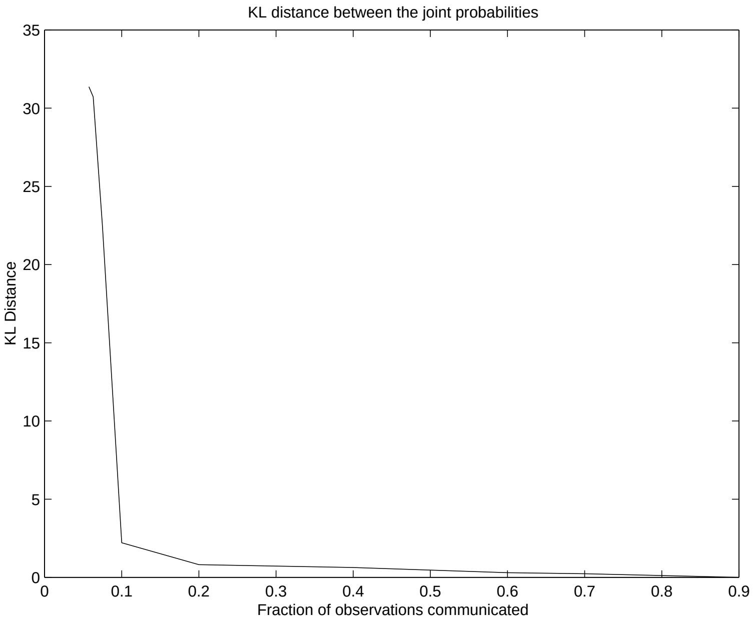

Fig. 6. KL distance between joint probabilities

is shown in figure 7. Each nodes takes discrete values, but not necessarily binary. The node variables of the ALARM network is as follows: 1-Anaphylaxis, 2-Intubation, 3-KinkedTube, 4-Disconnect, 5-MinVolSet, 6-VentMach,7-VentTube, 8-VentLung, 9-VentAlv, 10-ArtCO2, 11-TPR, 12-Hypovolemia, 13-Lvfailure, 14-StrokeVolume, 15-InsuffAnesth, 16-Pulm-Embolus, 17-Shunt, 18-FiO2, 19-PVSat, 20-SaO2, 21-Catechol, 22-HR, 23-CO, 24-BP, 25-LVEDVolume, 26-CVP, 27-ErrCauter, 28-ErrLowOutput, 29-ExpCO2, 30-HRBP, 31-HREKG, 32-HRSat, 33-History, 34-MinVol, 35-PAP, 36-PCWP, 37-Press.

In order to test the scalability of our approach with respect to number of nodes and observations, a dataset with 15000 samples was generated. These 37 nodes were split into 4 sites as follows – site 1: {3, 4, 5, 6, 7, 8, 15}, site2: {2, 9, 10, 18, 19, 29, 29, 34, 37}, site3: {16, 17, 20, 21, 22, 27, 30, 31, 32, 35}, and site4: {1, 11, 12, 13, 14, 23, 24, 25, 26, 28, 33, 36}. Note there are 13 cross edges.

We assumed that the structure of the Bayesian network was given, and tested our approach for estimating the conditional probabilities. The KL distance between the conditional probabilities (see Equation (21)) estimated based on our collective BN and a BN obtained using a centralized approach was computed. In particular, we illustrate the results for the conditional probabilities at two different nodes: 20, 21, both of which are cross terms. Figure 8 (left) depicts the CKL distance of node 20 between the two estimates. Figure 8 (right) depicts a similar CKL distance for node 21. Clearly, even with a small data communication, the estimates of the conditional probabilities of the cross-terms, based on our collective approach, is quite close to that obtained by the centralized approach. Note that node 21 has 4 parents — one of them being local (in the same site as node 21) with the other three being in a different sites. Also the conditional probability table of node 21 has 54 parameters, corresponding to the possible configurations

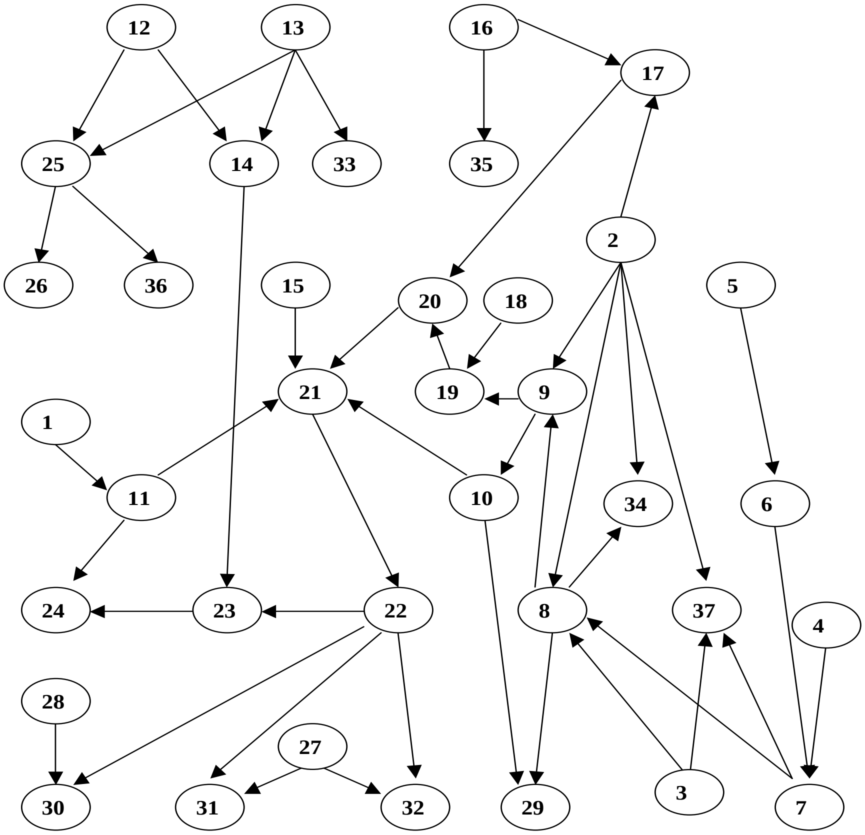

Fig. 7. ALARM Network

of node 21 and its four parents. Consequently, learning the parameters of this node is a non-trivial task. Our experiments clearly demonstrates that our technique can be used to perform a biased sampling for discovering relationships between variables across sites. This simulation also illustrates the fact that the proposed approach scales well with respect to number of nodes and samples.

Next, we test the scalability of our approach with respect to number of sites. As the number of sites increases, there are more cross edges. This means that, there are more nodes, whose parameters are to be learnt at the central site, using a limited portion of the data. Moreover, distribution of variables among more sites would affect the local models, likelihood computation at local sites, and consequently the samples selected for transmission. We successively split the variables across two, three, four, and five sites and in each case, illustrate the performance using the conditional probability of two fixed — nodes 20 and 21. Table 8 depicts the location of the parents of nodes 20 and 21 in each case (nodes 20 and 21 were always assigned to site 1).

By increasing the number of sites and distributing the parents of node 21 among more sites, the learning problem has been deliberately made more difficult. Figure 9 depicts the CKL distance for nodes 20 and 21, for the different cases. Clearly, the increasing of number of sites does make the collective learning

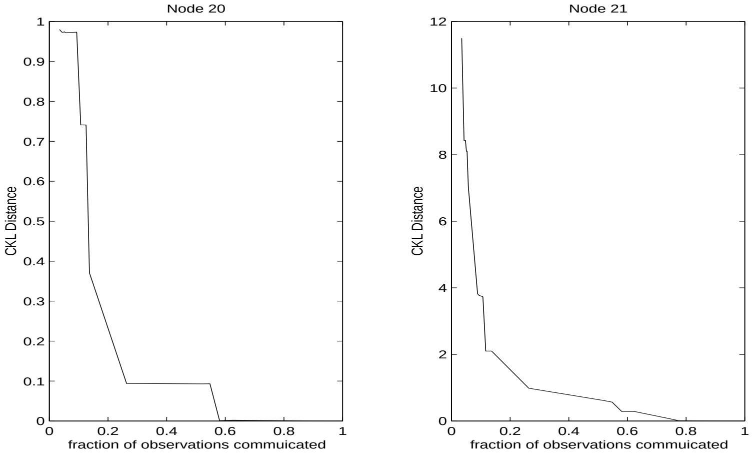

Fig. 8. KL distance between conditional probabilities for ALARM network

	parents of n20	parents of n21
2 sites	n16: site 1, n2: site 2	n20: site 1, n11, n15, n29: site 2
3 sites	n16: site 1, n2: site 2	n20: site 1, n11, n15: site 2, n29: site 3
4 sites	n16: site 1, n2: site 2	n20: site 1, n11: site 2, n15: site 3, n29: site 4
5 sites	n16: site 1, n2: site 2	n20: site 1, n11: site 2, n15: site 3, n29: site 4

Table 8. Location of parents of nodes 20, 21 for different cases

more difficult and the performance with smaller number of sites is better than that with larger number of sites. However, the CKL distance decreases rapidly even for large number of sites. Moreover, the performance of our approach is similar with increasing number of sites, after a relative small portion of samples are transmitted (from figure 9, about 35% samples transmitted). This clearly illustrates that our approach scales well with respect to number of sites.

5. Discussions and Conclusions

We have presented an approach to learning BNs from distributed heterogenous data. This is based on a collective learning strategy, where a local model is obtained at each site and the global associations are determined by a selective transmission of data to a central site. In our experiments, the performance of the collective BN was quite comparable to that obtained from a centralized approach, even for a small data communication. To our knowledge, this is the first approach to learning BNs from distributed heterogenous data.

Our experiments suggest that the collective learning scales well with respect to number of sites, samples, and features. Many interesting applications are possible from a BN model of the web log data. For example, specific structures in the overall BN would indicate special user patterns. This could be used to identify new user patterns and accordingly personalize offers and services pro-

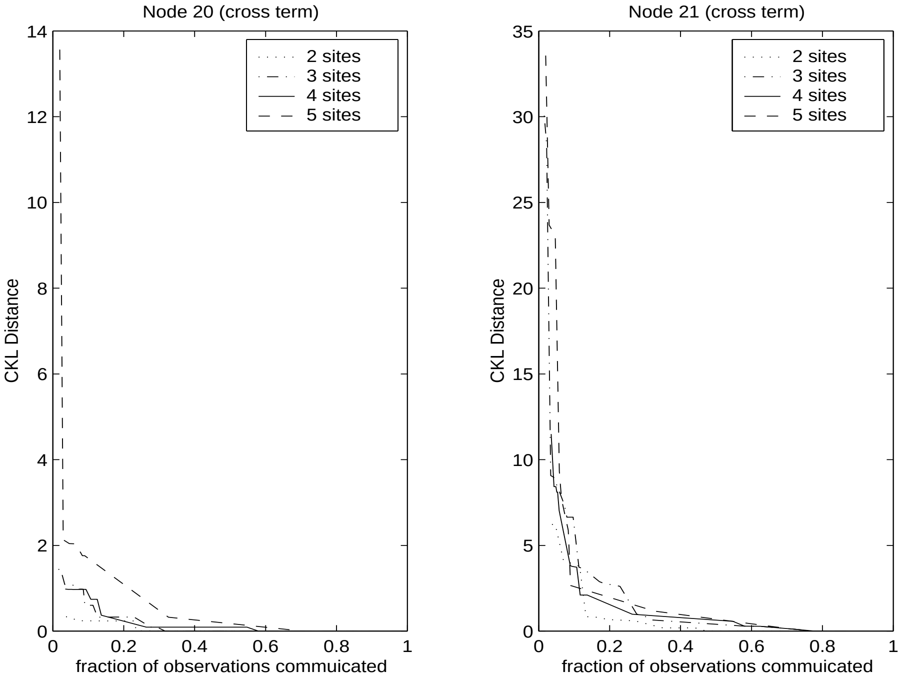

Fig. 9. KL distance between conditional probabilities for ALARM network for different number of sites

vided to such users. Another interesting application is to classify the users into different groups based on their usage patterns. This can be thought of decomposing the overall BN (obtained from the log data by collective learning) into a number of sub-BNs, each sub-BN representing a specific group having similar preferences. We are actively pursuing these ideas and would report results in a future publication.

We now discuss some limitations of our proposed approach, which suggest possible directions for future work.

- **Hidden node at local sites**: For certain network structures, it may not be possible to obtain the correct (local) links, based on local data at that site. For example, consider the ASIA model shown in Figure 1, where the observations corresponding to variables A, T, E, and X are available at site A and those corresponding to variables S, L, B, and D are available at site B. In this case, when we learn a local BN at site B, we would expect a (false) edge from node L to node D, because of the edges $L \rightarrow E$ and $E \rightarrow D$ in the overall BN and the fact that node E is "hidden" (unobserved) at site B. This was verified experimentally as well. However, the cross-links $L \rightarrow E$ and $E \rightarrow D$ were still detected correctly at the central site, using our "selectively sampled" data. Therefore, it is necessary to re-examine the local links after discovering the cross-links. In other words, some post-processing of the resulting overall BN is required to eliminate such false local edges. This can be done by evaluating an appropriate score metric on BN configurations with and without such suspect local links. We are currently pursuing this issue. Note, however, that we do not encounter this problem in the examples presented in Section 4.

- **Assumptions about the data**: As mentioned earlier, we assume the existence of a key that links observations across sites. Moreover, we consider a simple heterogenous partition of data, where the variable set at different sites are non-overlapping. We also assume that our data is stationary (all data points come from the same distribution) and free of outliers. These are simplifying assumptions to derive a reasonable algorithm for distributed Bayesian learning. Suitable learning strategies that would allow us to relax of some of these assumptions would be an important area of research.
- **Structure Learning**: Even when the data is centralized, learning the structure of BN is considerably more involved than estimating the parameters or probabilities associated with the network. In a distributed data scenario, the problem of obtaining the correct network structure is even more pronounced. The "hidden node" problem discussed earlier is one example of this. As in the centralized case, prior domain knowledge at each local site, in the form of probabilistic independence or direct causation, would be very helpful. Our experiments on the ASIA model demonstrate that the proposed collective BN learning approach to obtain the network structure is reasonable, at least for simple cases. However, this is just a beginning and deserves careful investigation.
- **Parameter Learning**: The proposed collective method is designed for structure and parameter learning. The likelihood computation in local site is not a trivial job and introduces some computational overhead. This may not be acceptable for real-time applications such as online monitoring of stock market data. For real time or online learning applications, we need to considerably reduce the amount of computation at the local sites. Towards that end, we have proposed a new collective learning method (Chen and Sivakumar, 2002) for learning the parameters of a BN (here we assumes that the structure of the BN is known) which dramatically reduces the local computation time.
- **Performance Bounds**: Our approach to "selective sampling" of data that maybe evidence of cross-terms is reasonable based on the discussion in Section 3 (see eq. (4)-(7)). This was verified experimentally for the three examples in Section 4. Currently, we are working towards obtaining bounds for the performance of our collective BN as compared to that obtained from a centralized approach, as a function of the data communication involved.

Acknowledgements. This work was partially supported by NASA, under Cooperative agreement NCC 2-1252. We would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

Abe, N., Takeuchi, J. and Warmuth, M. (1991), Polynomial learnability of probabilistic concepts with respect to the Kullback-Leibler divergence, *in* 'Proceedings of the 1991 Workshop on Computational Learning Theory', pp. 277–289.

Bauer, E., Koller, D. and Singer, Y. (1997), Update rules for parameter estimation in Bayesian networks, *in* D. Geiger and P. Shanoy, eds, 'Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann, pp. 3–13.

Beinlich, I., Suermondt, H., Chavez, R. and Cooper, G. (1989), The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks, *in* 'Proceedings of the Second European Conference on Artificial Intelligence in Medical Care', Springer-Verlag, pp. 247–256.

Binder, J., Koller, D., Russel, S. and Kanazawa, K. (1997), 'Adaptive probabilistic networks with hidden variables', *Machine Learning* **29**, 213–244.

Bouckaert, R. R. (1994), Properties of Bayesian network learning algorithms, *in* R. L. de Mantaras and D. Poole, eds, 'Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann, pp. 102–109.

Breese, J. S., Heckerman, D. and Kadie, C. (1998), Empirical analysis of predictive algorithms for collaborativefiltering, *in* G. F. Cooper and S. Moral, eds, 'Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann.

Buntine, W. (1991), Theory refinement on Bayesian networks, *in* B. D. D'Ambrosio and P. S. amd P. P. Bonissone, eds, 'Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann, pp. 52–60.

Cheeseman, P. and Stutz, J. (1996), Bayesian classification (autoclass): Theory and results, *in* U. Fayyad, G. P. Shapiro, P. Smyth and R. S. Uthurasamy, eds, 'Advances in Knowledge Discovery and Data Mining', AAAI Press.

Chen, R. and Sivakumar, K. (2002), A new algorithm for learning parameters of a Bayesian network from distributed data, *in* '(To be presented) Proceedings of the 2002 IEEE International Conference on Data Mining', Japan.

Chen, R., Sivakumar, K. and Kargupta, H. (2001*a*), An approach to online Bayesian learning from multiple data streams, *in* H. Hargupta, K. Sivakumar and R. Wirth, eds, 'Proceedings of the Workshop on Ubiquitous Data Mining: Technology for Mobile and Distributed KDD (In the 5th European Conference, PKDD 2001)', Freiburg, Germany, pp. 31–45.

Chen, R., Sivakumar, K. and Kargupta, H. (2001*b*), Distributed web mining using Bayesian networks from multiple data streams, *in* 'Proceedings of the 2001 IEEE International Conference on Data Mining', San Jose, CA.

Cheng, J., Bell, D. A. and Liu, W. (1997), Learning belief networks from data: An information theory based approach, *in* 'Proceedings of the Sixth ACM International Conference on Information and Knowledge Management'.

Chickering, D. M. (1996), Learning equivalence classes of Bayesian network structure, *in* E. Horvitz and F. Jensen, eds, 'Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann.

Chickering, D. M. and Heckerman, D. (1997), 'Efficient approximation for the marginal likelihood of incomplete data given a Bayesian network', *Machine Learning* **29**, 181–212.

Cooper, G. F. and Herskovits, E. (1992), 'A Bayesian method for the induction of probabilistic networks from data', *Machine Learning* **9**, 309–347.

Dasgupta, S. (1997), 'The sample complexity of learning fixed-structure Bayesian networks', *Machine Learning* **29**, 165–180.

Ezawa, K. J. and T, S. (1995), Fraud/uncollectable debt detection using Bayesian network based learning system: A rare binary outcome with mixed data structures, *in* P. Besnard and S. Hanks, eds, 'Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann, pp. 157–166.

Friedman, N. (1998), The Bayesian structural EM algorithm, *in* G. F. Cooper and S. Moral, eds, 'Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann.

Friedman, N., Geiger, D. and Goldszmidt, M. (1997), 'Bayesian network classifiers', *Machine Learning* **29**, 131–163.

Friedman, N. and Goldszmidt, M. (1997), Sequential update of Bayesian network structure, *in* D. Geiger and P. Shanoy, eds, 'Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann.

Friedman, N. and Yakhini, Z. (1996), On the sample complexity of learning Bayesian networks, *in* 'Proceedings of the twelfth conference on uncertainty in artificial intelligence'.

Gilks, W., Richardson, S. and Spiegelhalter, D. (1996), *Markov chain Monte Carlo in practice*, Chapman and Hall.

Heckerman, D. (1998), A tutorial on learning with Bayesian networks, *in* M. I. Jordan, ed., 'Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models', Kluwer Academic Publishers.

Heckerman, D., Geiger, D. and Chickering, D. M. (1995), 'Learning Bayesian networks: The combination of knowledge and statistical data', *Machine Learning* **20**, 197–243.

Heckerman, D. and Gieger, D. (1995), Learning Bayesian networks: A unification for discrete and Gaussian domains, *in* P. Besnard and S. Hanks, eds, ‘Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence’, Morgan Kaufmann, pp. 274–284.

Hershberger, D. and Kargupta, H. (2001), ‘Distributed multivariate regression using wavelet-based collective data mining’, *Journal of Parallel and Distributed Computing* **61**, 372–400.

Jensen, F. (1996), *An Introduction to Bayesian Networks*, Springer.

Kargupta, H., Huang, W., Sivakumar, K. and Johnson, E. (2001), ‘Distributed clustering using collective principal component analysis’, *Knowledge and Information Systems Journal* **3**, 422–448.

Kargupta, H., Park, B., Hershberger, D. and Johnson, E. (2000), Collective data mining: A new perspective toward distributed data mining, *in* H. Kargupta and P. Chan, eds, ‘Advances in Distributed and Parallel Knowledge Discovery’, AAAI/ MIT Press, Menlo Park, California, USA, pp. 133–184.

Kenji, Y. (1997), Distributed cooperative Bayesian learning strategies, *in* ‘Proceedings of the Tenth Annual Conference on Computational Learning Theory’, ACM Press, Nashville, Tennessee, pp. 250–262.

Lam, W. and Bacchus, F. (1994), ‘Learning Bayesian belief networks: An approach based on the MDL principle’, *Computational Intelligence* **10**, 262–293.

Lauritzen, S. L. (1995), ‘The EM algorithm for graphical association models with missing data’, *Computational Statistics and Data Analysis* **19**, 191–201.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988), ‘Local computations with probabilities on graphical structures and their application to expert systems (with discussion)’, *Journal of the Royal Statistical Society, series B* **50**, 157–224.

Madigan, D. and Raftery, A. (1994), ‘Model selection and accounting for model uncertainty in graphical models using Occam’s window’, *Journal of the American Statistical Association* **89**, 1535–1546.

McLean, B., Hawkins, C., Spagna, A., Lattanzi, M., Lasker, B., Jenkner, H. and White, R. (1998), ‘New horizons from multi-wavelength sky surveys’, *IAU Symposium. 179* .

Meila, M. and Jordan, M. I. (1998), Estimating dependency structure as a hidden variable, *in* ‘NIPS’.

Park, B. and Kargupta, H. (2002), Distributed data mining: Algorithms, systems, and applications, *in* N. Ye, ed., ‘(To appear) Data Mining Handbook’.

Pearl, J. (1993), ‘Graphical models. causality and intervention’, *Statistical Science* **8**, 266–273.

Singh, M. (1997), Learning Bayesian networks from incomplete data, *in* ‘Proceedings of the National Conference on Artificial Intelligence’, AAAI Press, pp. 27–31.

Singh, M. and Provan, G. M. (1995), A comparison of induction algorithms for selective and non-selective Bayesian classifiers, *in* A. Prieditis and S. Russel, eds, ‘Proceedings of the Twelfth International Conference on Machine Learning’, Morgan Kaufmann, pp. 497–505.

Spiegelhalter, D. J. and Lauritzen, S. L. (1990), ‘Sequential updating of conditional probabilities on directed graphical structures’, *Networks* **20**, 570–605.

Spirtes, P., Glymour, C. and Scheines, R. (1993), *Causation, Prediction and Search*, number 81 *in* ‘Lecture Notes in Statistics’, Springer-Verlag.

Suzuki, J. (1993), A construction of Bayesian networks from databases based on an MDL scheme, *in* D. Heckerman and A. Mamdani, eds, ‘Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence’, Morgam Kaufmann, pp. 266–273.

Szalay, A. (1998), ‘The evolving universe’, *ASSL* (231).

Thiesson, B. (1995), Accelerated quantification of Bayesian networks with incomplete data, *in* ‘Proceedings of the First International Conference on Knowledge Discovery and Data Mining’, AAAI Press, pp. 306–311.

Thomas, A., Spiegelhalter, D. and Gilks, W. (1992), Bugs: A program to perform Bayesian inference using Gibbs sampling, *in* J. Bernardo, J. Berger, A. Dawid and A. Smith, eds, ‘Bayesian Statistics’, Oxford University Press, pp. 837–842.

Zweig, G. and Russel, S. J. (1998), Speech recognition with dynamic Bayesian networks, *in* ‘Proceedings of the Fifteenth National Conference on Artificial Intelligence’.

Author Biographies

insert photo

Rong Chen received a B.E. degree from Southeast University, Nanjing, China, in 1996 and an M.S. degree from the Graduate School of Chinese Academy of Science, Beijing, China, in 1999. He is currently a Ph.D. student at the School of Electrical Engineering and Computer Science, Washington State University, WA. His research interests include distributed data mining, Bayesian network, and statistical image model.

insert photo

Krishnamoorthy Sivakumar is an Assistant Professor in the School of Electrical Engineering and Computer Science, Washington State University. His current research interests include statistical signal processing, Markov models and Bayesian inference for images, and application of signal processing techniques to knowledge discovery and distributed data mining. He received an M.S. in Mathematical Sciences and M.S. and Ph.D. in Electrical and Computer Engineering, in 1995, 1993, and 1997, respectively, all from The Johns Hopkins University. He has published more than thirty technical articles in journals, international conferences, and book chapters, and has one pending patent application. He is a member of the editorial board of the Journal of mathematical imaging and vision and a co-chair of the Workshop on Ubiquitous Data Mining for Mobile and Distributed Environments in the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)

.

insert photo

Hillol Kargupta is an Assistant Professor at the Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County. He received his Ph.D. in Computer Science from University of Illinois at Urbana-Champaign in 1996. His research interests include mobile and distributed data mining, and computation in gene expression. Dr. Kargupta won the National Science Foundation CARRER award in 2001 for his research on ubiquitous and distributed data mining. He won the 1997 Los Alamos Award for Outstanding Technical Achievement. His dissertation earned him the 1996 Society for Industrial and Applied Mathematics (SIAM) annual best student paper prize. He has published more than sixty peer-reviewed articles in journals, conferences, and books. He is the primary editor of a book entitled "Advances in Distributed and Parallel Knowledge Discovery", AAAI/MIT Press. He is an Associate Editor of the IEEE Transactions on Systems, Man, Cybernetics, Part B. He is in the organizing committee of the 2001, 2002, 2003 SIAM Data Mining Conference, organizing/program committee of the 2003 and 2001 ACM SIGKDD Conference, program committee of 2002 IEEE Conference on Data Mining, among others. He has organized numerous workshops and journal special issues. More information about him can be found at `http://www.cs.umbc.edu/~hillol`

.

Correspondence and offprint requests to: K. Sivakumar, School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164-2752, USA. Email siva@eecs.wsu.edu